

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

REMARKS

Claim 70 has been amended. Claims 1-70 remain in the application for consideration. In view of the following amendments and remarks, Applicant respectfully solicits allowance of the application and furtherance onto issuance.

Applicant has attempted to advance prosecution in this matter. Unfortunately, it appears that Applicant and the Office have reached a stalemate in prosecution. Applicant sincerely believes that the subject matter of the claims, in their present state, recite patentable subject matter over the references that have been cited by the Office. In the event that Applicant and the Office cannot reach an agreement after the filing of this response, Applicant intends to file a Notice of Appeal. As such, Applicant would respectfully request for the Office to contact the undersigned in the event the Office disagrees with Applicant's position, for the purpose of attempting to avoid the time and expense associated with an appeal.

New Claim 70

Claim 70 has been added. Applicant wishes for the record to reflect that this claim is not added in view of any patentability issues pertaining to the prior art cited by the Office.

§ 103(a) Rejections

Claims 1-69 stand rejected under 35 U.S.C § 103(a), as being obvious over U.S. Patent No. 6,253,366 to Mutschler, III (hereinafter "Mutschler") in view of U.S. Patent Application Publication No. 2002/0032768 to Voskuil.

The §103 Standard

In making out a §103 rejection, the Federal Circuit has stated that when one or more reference or source of prior art is required in establishing obviousness, “it is necessary to ascertain whether the prior art *teachings* would appear to be sufficient to one of ordinary skill in the art to suggest making the claimed substitutions or other modification.” *In re Fine*, 5 USPQ 2d, 1596, 1598 (Fed. Cir. 1988). That is, to make out a *prima facie* case of obviousness, the references must be examined to ascertain whether the combined *teachings* render the claimed subject matter obvious. *In re Wood*, 202 USPQ 171, 174 (C.C.P.A. 1979).

Moreover, there is a requirement that there must be some reason, suggestion, or motivation *from the prior art*, as a whole, for the person of ordinary skill to have combined or modified the references. *See, In re Geiger*, 2 USPQ 2d 1276, 1278 (Fed. Cir. 1987). It is impermissible to use the claimed invention as an instruction manual or “template” to piece together the teachings of the prior art so that the claimed invention is rendered obvious. One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention. *In re Fritch*, 23 USPQ 2d 1780, 1784 (Fed. Cir. 1992).

A factor cutting against a finding of motivation to combine or modify the prior art is when the prior art *teaches away* from the claimed combination. A reference is said to teach away when a person of ordinary skill, upon reading the reference, would be led in a direction divergent from the path that the applicant took. *In re Gurley*, 31 USPQ 2d 1130, 1131 (Fed. Cir. 1994).

1 In order for a *prima facie* case of obviousness to be made, the resulting
2 combination or motivation must appear to show or suggest the claimed invention.
3 *In re Nielson*, 2 USPQ 2d1525, 1528 (Fed. Cir. 1987).

4 5 The Mutschler Reference

6 Mutschler discloses a method and system for generating a compact
7 document type definition for data interchange among software tools.

8 As instructed in Mutschler's "Background" section, repositories provide a
9 central place for recording metadata and enable one to store, manage, share and
10 reuse *information about data* (i.e., metadata) that an enterprise uses. A repository
11 can store definitional, management and operational information. Tools can be
12 integrated with the repository to support information sharing and metadata reuse
13 and tool and technology models may be developed to manipulate the tool
14 information in the repository. *However, the transferring of data within models*
15 *from tool to tool or from a tool to the repository has been a cumbersome and*
16 *unyielding task for a long time.*

17 Repository models typically contain, as instructed by Mutschler, classes,
18 datatypes and messages. As more and more complex models are being built, the
19 need arises for a method and system to transfer data in a model from place to
20 place, e.g., to a tool that understands the UML ("Universal Modeling Language").
21 Mutschler instructs that his invention solves this problem by generating a data-
22 transfer syntax in which a tool using a meta-model can transport data from place to
23 place.

24 Mutschler instructs that the prefix "meta" as used in its patent shall describe
25 a relationship. For example, "meta-data" describes data. In a similar fashion, a

1 meta-object is an object that represents "meta-data"; and, "meta-model" means a
2 model that defines an abstract language for expressing other models. A "meta-
3 metamodel" means a model that defines an abstract language for expressing meta-
4 models. The relationship between a meta-metamodel and a meta-model is
5 analogous to the relationship between a meta-model and a model.

6 As instructed by Mutschler, it is a tedious and time consuming task to
7 generate a format description for enabling the interchange of metadata among
8 repositories and each different type of modeling tool available. Accordingly, there
9 is a need for automatically generating *format descriptions to expedite interchange*
10 *of metadata* among repositories and modeling tools.

11 Mutschler's solution automates the production of an XML DTD for meta-
12 models stored in a MOF-compliant repository by using the XML entity, which
13 allows one to group various Attributes, Associations and Compositions for
14 definitional reference in a meta object framework.

15 As instructed by Mutschler, starting in column 4, line 19, users of
16 workgroup-based and component development tools are finding it increasingly
17 difficult to coordinate their software development efforts across the enterprise. A
18 solution in accordance with the present invention employs the benefits of XMI
19 (XML Metadata Interchange), which is an open industry standard that combines
20 the benefits of the Web-based XML standard for defining, validating and sharing
21 document formats on the Web with the Meta Object Framework (MOF) *to provide*
22 *a means for generating formats to allow the development tools to share*
23 *information*. One particular use of this invention is to define an XML DTD for the
24 object-oriented Unified Modeling Language (UML). The XMI specification
25 provides application developers with a common language for specifying transfer

1 syntax for development language that allows visualizing, constructing and
2 documenting of distributed objects and business models. The XMI specification in
3 conjunction with the present invention will enable integration of development
4 tools from multiple vendors, collaboration and distribution of object-oriented
5 design and database schema information, and enhancement of the life cycle of
6 information resources.

7 As further instructed by Mutschler, software architectures based on meta-
8 models are transforming how one can design an architecture for dynamic
9 distributed systems. The UML and MOF specifications establish a robust meta-
10 model architecture for distributed object applications. The XMI specification
11 extends this architecture by leveraging technologies developed for the Web to
12 exchange models between tools, applications and repositories.

13 In order to accomplish the objects of the present invention, Mutschler
14 instructs that it is necessary to generate Document Type Definitions ("DTD") for
15 the Extensible Markup Language ("XML"), a World Wide Web Consortium
16 standard. A DTD is a set of rules governing the element types that are allowed
17 within an XML document and rules specifying the allowed content and attributes
18 of each element type. The DTD also declares all the external entities referenced
19 within the document and the notations that can be used. Stated otherwise, an XML
20 DTD provides a means by which an XML processor can validate the syntax and
21 some of the semantics of an XML document. An XMI DTD specifies the
22 particular elements allowed in an XMI document.

23 Mutschler notes that his invention describes algorithms for generating an
24 XMI DTD for any valid meta-model defined in a MOF-compliant repository. The
25 material starting in column 4, line 63 describes how this is done.

1 Nowhere does Mutschler describe *software extensions*, as that term is
2 utilized in Applicant's claims and defined in Applicant's specification. The Office
3 appears to argue that such is disclosed in Mutschler, particularly at column 4, lines
4 21-30 and 48-60.

5 A careful reading of these sections of Mutschler, however, indicates that
6 these sections do not describe *software extensions*, as that term is utilized in
7 connection with Applicant's disclosure.

8 9 **The Voskuil Reference**

10 Voskuil discloses a system and a method for configuring remotely located
11 software applications and/or application components. Specifically, Voskuil's
12 systems and methods are disclosed to discover the presence and configuration of
13 one or more remotely located software applications and/or components and for
14 determine and making changes to the configuration.

15 According to Voskuil, its system includes an end user computer connected
16 to an autopfile server that can send information to and receive information from
17 the end user computer. An application on the end user computer communicates
18 with the autopfile server and can receive and execute extension modules that
19 expand the functionality of the application. The autopfile server can include a
20 repository of extension modules that are adapted for detecting the installation of
21 one or more applications on the end user computer and reporting the identity of the
22 detected applications and their configuration information to the autopfile server.
23 The autopfile server can use this information to prompt the end user to select the
24 application or applications that are to be automatically configured. The autopfile
25 server can then use the information concerning the detected applications and their

1 existing configuration to select the *configuration instructions* to be sent to the end
2 user computer to automatically configure each installed application according to
3 the end user and/or third party requirements. See, e.g., paragraph [0012].

4 According to Voskuil, its method includes the steps of a remotely located
5 end user computer establishing a communication session with a service provider
6 server. The autopfile server transfers a *discovery module* and/or discovery
7 information to the end user computer. The end user computer executes the
8 discovery module to generate data representative of the installed applications,
9 application components, and/or application configurations on the end user
10 computer. This data is then transferred to the autopfile server. The end user is
11 then queried to indicate which installed applications and/or application
12 components the user desires configured. For each installed application and/or
13 application component, the data is analyzed to determine the proper configuration
14 for each. A configuration module and/or configuration information is transferred
15 to the end user computer and executed to configure each installed application
16 and/or application component selected by the end user. Data representative of the
17 results of configuration of each application and/or application component are
18 generated and transferred and the results of the configuration are reported to the
19 end user. The results are then logged on the autopfile server for access by the
20 service provider. See, e.g. paragraph [0013].

21 As instructed by Voskuil, the end user computer can include a client
22 application that is used to communicate with the autopfile server. The client
23 application can have the ability to be extended by external or third party software
24 components or modules. The autopfile server can include an autopfile
25 extension, a client application extension that can be transmitted to and installed on

1 the end user computer. Once the autopfile extension is installed on the end user
2 computer, the autopfile extension can send information to and receive
3 autopfile information from the autopfile server. See, e.g., paragraph [0014].

4 Further on in Voskuil's disclosure, Voskuil describes how its system works.
5 Specifically, starting at around paragraph [0027], Voskuil instructs as follows.

6 Remote devices 110 (see Fig. 1) can utilize any application that can
7 communicate with the autopfile server 130 over the network 140. In addition to
8 enabling the remote devices 110 to communicate with remote servers, the client
9 application should include the ability to execute plug-ins or extensions to the client
10 application, such as for example, Microsoft Active-X controls and/or Netscape
11 plug-ins, that are capable of creating, and reading from and writing to files on the
12 remote device 110 or the ability to spawn processes that are capable of creating,
13 and reading from and writing to files on the remote device 110.

14 In one embodiment, the plug-ins or extensions can be transferred from the
15 autopfile server 130 to the remote device 110 prior to execution. Alternatively,
16 the plug-ins or extensions can be included with the client application or may be
17 implemented as standalone applications having the ability to read from and write
18 to files with their own communications capability.

19 FIG. 2 shows a diagrammatic view of a system 200 for determining which
20 applications and/or application components are installed on a remote PC and for
21 configuring one or more of the applications and/or application components in
22 accordance with the present invention. In one embodiment, the system 200
23 includes a client operating system 210 installed on one end user PC connected to a
24 remotely located server 230 via a network 240 such as a TCP/IP network or the
25 Internet. A client application 212, such as Netscape Navigator or Microsoft

1 Internet Explorer, is used to communicate with a remotely located profile server
2 230 over the network 240 via HTTP.

3 The client application 212 supports an extension interface such as Netscape
4 Navigator's plug-in interface 214, Microsoft Internet Explorer's Active-X controls
5 214 or Sun Microsystems Java 214 that allow the client application 212 to be
6 extended to permit additional functionality. The extension interface 214 permits
7 extension modules 216 (e.g. plug-ins or Active-X controls) to access the file
8 system to create, read and write to files on the remote PC. Alternatively, the
9 extension interface 214 can permit the spawning of a process that can access the
10 file system to create, read and write to files on the end user PC. In the illustrative
11 embodiment, this can be accomplished via either a Netscape plug-in or a
12 Microsoft Active-X control that includes one or more modules 216 common
13 functionality that permit the extension module to access the file system as
14 necessary to detect create and modify files in order to automatically configure the
15 remote PC. The extension modules 216 can be adapted to receive *instructions*
16 *from the profile server 230 regarding the files to be examined or modified and*
17 *specific modifications to be made.* In one embodiment, the *instructions* can be
18 incorporated in an XML data structure 246 that is transmitted from the server 230.
19 *The XML instructions dictate which modules and functions are executed, the*
20 *parameters for each, and how the results are processed.* Alternatively, the
21 extension modules 216 can be preprogrammed to perform some or all of the
22 necessary tasks to discover and/or configure applications on the remote PC,
23 without receiving instructions from the server 230.

24 As instructed by Voskuil, the profile server 230 can transfer the XML
25 instructions to the remote PC using the same data channel with which the

1 autoprofile server 230 communicates with the client application guaranteeing that
2 the communications channel is valid. Secure communications channels can be
3 provided using well know protocols such as Secure Sockets and HTTPS.

4 5 **Claims 1-16**

6 **Claim 1** recites a method comprising [emphasis added]:

- 7
- 8 • describing one or more software extensions using descriptions, the
9 extensions being configured for incorporation in a software platform
10 executing on a client; and
 - 11 • delivering the descriptions of the one or more extensions to the client
12 via a network, the descriptions being configured for use in
13 downloading the software extensions via the network;
 - 14 • said acts of describing and delivering being configured to enable
15 *software* to be delivered over the network.

16
17 In making out the rejection of this claim, the Office argues that its subject
18 matter is obvious over Mutschler in view of Voskuil. As pointed out above,
19 Mutschler has nothing to do with software extensions as that term is utilized in this
20 claim. Rather, Mutschler pertains to tools that allow data interchange among
21 software tools. The Office cites to various sections of Mutschler including column
22 1, lines 29-32; column 2, lines 37-39; column 4, lines 25-26; column 4, lines 35-39
23 in support of its position. Yet, these excerpts have nothing to do with software
24 extensions or their delivery.

25 The Office then relies on Voskuil and argues that it discloses delivering the
descriptions of the one or more extensions to the client via a network, the
descriptions being configured for use in downloading the software extensions via
the network, citing to paragraph [0014] and [0029] for support.

These excerpts are reproduced below for the convenience of the Office:

1 [0014] The end user computer can include a client application that is
2 used to communicate with the autoprofile server. The client application can
3 have the ability to be extended by external or third party software
4 components or modules. The autoprofile server can include an autoprofile
5 extension, a client application extension that can be transmitted to and
6 installed on the end user computer. Once the autoprofile extension is
7 installed on the end user computer, the autoprofile extension can send
8 information to and receive autoprofile information from the autoprofile
9 server.

10 [0029] The client application 212 supports an extension interface
11 such as Netscape Navigator's plug-in interface 214, Microsoft Internet
12 Explorer's Active-X controls 214 or Sun Microsystems Java 214 that allow
13 the client application 212 to be extended to permit additional functionality.
14 The extension interface 214 permits extension modules 216 (e.g. plug-ins
15 or Active-X controls) to access the file system to create, read and write to
16 files on the remote PC. Alternatively, the extension interface 214 can
17 permit the spawning of a process that can access the file system to create,
18 read and write to files on the end user PC. In the illustrative embodiment,
19 this can be accomplished via either a Netscape plug-in or a Microsoft
20 Active-X control that includes one or more modules 216 common
21 functionality that permit the extension module to access the file system as
22 necessary to detect create and modify files in order to automatically
23 configure the remote PC. The extension modules 216 can be adapted to
24 receive instructions from the profile server 230 regarding the files to be
25 examined or modified and specific modifications to be made. In one
embodiment, the instructions can be incorporated in an XML data structure
246 that is transmitted from the server 230. The XML instructions dictate
which modules and functions are executed, the parameters for each, and
how the results are processed. Alternatively, the extension modules 216 can
be preprogrammed to perform some or all of the necessary tasks to discover
and/or configure applications on the remote PC, without receiving
instructions from the server 230.

22 Applicant has reviewed Voskuil generally, and these excerpts specifically,
23 and can find no disclosure of *delivering descriptions of one or more extensions to*
24
25

1 a client, where the descriptions are configured for use in downloading the
2 software extensions.

3 If the Office continues to maintain its position with regard to Voskuil
4 disclosing this subject matter, then Applicant respectfully requests the Office to
5 specifically identify for Applicant what it considers to be a “description” so as to
6 supply this element of the recited claim.

7 In addition, the Office argues as motivation to combine these two
8 references as follows: “...because Mutschler’s code for object models or code for
9 software extensions, as disclosed by Voskuil, is merely code.” See, Office Action,
10 page 4-5. Applicant fails to fully understand or appreciate this motivation and
11 respectfully submits that it does not rise to the level of particular findings
12 sufficient to support a *prima facie* case of obviousness.

13 Hence, for this additional reason, the Office has failed to establish a *prima*
14 *facie* case of obviousness and this claim is allowable.

15 **Claims 2-16** depend from claim 1 and are allowable as depending from an
16 allowable base claim. These claims are also allowable for their own recited
17 features which, in combination with those recited in claim 1, are neither disclosed
18 nor suggested in the references of record, either singly or in combination with one
19 another.

20
21 **Claim 17**

22 **Claim 17** recites one or more computer-readable media having computer-
23 readable instructions thereon which, when executed by a computer system, cause
24 the computer system to [emphasis added]:
25

- describe one or more software extensions using extensible markup language (XML), the extensions being configured for incorporation in a software platform comprising a single application program, the single application program having multiple different functionalities that can enable a user to accomplish multiple different tasks; and
- deliver XML descriptions of the one or more extensions to the client via the Internet, the descriptions being configured for use in downloading the software extensions via the Internet;
- wherein causing said computer system to describe one or more extensions and deliver XML descriptions enables *software* to be delivered over the Internet.

In making out the rejection of this claim, the Office argues that its subject matter is obvious over Mutschler in view of Voskuil. As pointed out above, Mutschler has nothing to do with software extensions as that term is utilized in this claim. Rather, Mutschler pertains to tools that allow data interchange among software tools. The excerpts of Mutschler cited by the Office have nothing to do with software extensions or their delivery.

Further, the Office's reliance on Voskuil appears to be misplaced. Applicant has reviewed Voskuil generally, and the excerpts relied upon by the Office specifically, and can find no disclosure of *delivering descriptions of one or more extensions to a client, where the descriptions are configured for use in downloading the software extensions.*

If the Office continues to maintain its position with regard to Voskuil disclosing this subject matter, then Applicant respectfully requests the Office to *specifically* identify for Applicant what it considers to be a "description" so as to supply this element of the recited claim.

Claims 18-28

Claim 18 recites a method comprising [emphasis added]:

- describing one or more software extensions using one or more descriptive files, the extensions being configured for incorporation in a software program executing on a client;
- associating the one or more descriptive files with one or more associated extension files that are useable to provide a program functionality;
- storing the descriptive files and associated extension files in a network-accessible location; and
- delivering the descriptive files and the associated extension files of the one or more extensions to the client via a network.

In making out the rejection of this claim, the Office argues that its subject matter is obvious over Mutschler in view of Voskuil. As pointed out above, Mutschler has nothing to do with software extensions as that term is utilized in this claim. Rather, Mutschler pertains to tools that allow data interchange among software tools. The excerpts of Mutschler cited by the Office have nothing to do with software extensions or their delivery.

Further, the Office argues that the claim's recited "descriptive files" is somehow met by Mutschler's XML DTD. See, e.g., Office Action, page 9. The Office is mistaken. The recited "descriptive files" describe one or more software extensions. As Mutschler does not disclose software extensions in the same context as utilized in this claim and defined in Applicant's specification, it is virtually impossible for Mutschler to disclose descriptive files that describe one or more of the software extensions. Mutschler's DTD simply refers to a set of rules governing the element types that are allowed within an XML document and rules specifying the allowed content and attribute types for each element type. See, column 4, lines 48-60.

1 Further, the Office argues that Voskuil discloses associating the one or
2 more descriptive files with one or more extension files that are useable to provide
3 a program functionality, citing to Voskuil's paragraph [0015]. The entirety of this
4 paragraph is set out below for the convenience of the Office:

5
6 [0015] The autopfile extension can include one or more
7 autopfile modules. Each module can include one or more functions that
8 can be used to access files on the end user computer, search for information
9 in a file, modify an existing file, or create a new file as necessary to
10 configure a given application. The autopfile modules can be instructed or
11 programmed to execute one or more of the library of functions by a list of
12 instructions received from the autopfile server. The autopfile discovery
13 information can be used to program the autopfile discovery module to
14 search for one or more installed applications, application components, or
15 application configurations. The autopfile server can use the results of the
16 autopfile discovery process, as well as end user and SP preferences, to
17 create configuration instructions. The autopfile configuration module
18 performs the configuration of applications or application components in
19 accordance with the configuration instructions and reports the results back
20 to the autopfile server.

21
22 This excerpt merely discusses the autopfile extension which, as indicated
23 in Voskuil's paragraph [0014], simply serves to send information to and receive
24 autopfile information from the autopfile server. Part of the process of being
25 able to send information to the autopfile server pertains to discovering such
information on the client device. Hence, as indicated in paragraph [0015], the
autopfile extension is imparted with functionality that allows it to access files on
the end users computer, search for information in a file, modify an existing file, or
create a new file to configure an application.

26
27 This excerpt in no way anticipates associating one or more descriptive files
28 with one of more associated extension files as recited in this claim. If the Office

1 disagrees, then Applicant respectfully requests the Office to specifically identify
2 what it considers to be Voskuil's "descriptive files" and what it considers to be
3 Voskuil's "associated extension files".

4 As motivation for combining these references, the Office argues that it
5 would be obvious to combine the reference because "both inventions provide for
6 code stored in a repository/database and delivered to a remote location using XML
7 using a description of the content, in the form of a DTD document." See Office
8 Action, page 10. Applicant respectfully submits that this motivation falls far short
9 of the particular findings that one must make in order to support a *prima facie* case
10 of obviousness.

11 Accordingly, for at least these reasons, the Office has failed to establish a
12 *prima facie* case of obviousness.

13 **Claims 19-28** depend from claim 18 and are allowable as depending from
14 an allowable base claim. These claims are also allowable for their own recited
15 features which, in combination with those recited in claim 18, are neither disclosed
16 nor suggested in the references of record, either singly or in combination with one
17 another.

18
19 **Claims 29-39**

20 **Claim 29** recites a method comprising [emphasis added]:

- 21
- 22 • storing one or more extension definition files (EDFs) that describe a
logical attachment to a software application program;
 - 23 • storing one or more extension files that correspond to the one or
more EDFs and extend the software application program;
 - 24 • delivering, via a network, at least one EDF to a client; and
 - 25 • delivering, via the network, at least one extension file that
corresponds to the at least one EDF to a client;

- both of said acts of storing and both of said acts of delivering enabling *software* to be delivered over the network.

In making out the rejection of this claim, the Office argues that its subject matter is obvious over Mutschler in view of Voskuil. As pointed out above, Mutschler has nothing to do with software extensions as that term is utilized in this claim. Rather, Mutschler pertains to tools that allow data interchange among software tools. The excerpts of Mutschler cited by the Office have nothing to do with software extensions or their delivery.

Specifically, the Office argues that Mutschler discloses storing one or more extension definition files (EDFS), citing to Figs. 1 and 2 and to Mutschler's DTD. A DTD simply does not anticipate an extension definition file as utilized in this claim and defined in Applicant's specification. Hence, for at least this reason, the Office has failed to establish a *prima facie* case of obviousness.

Further, the Office appears to rely on Voskuil for the remainder of the subject matter recited in this claim. Specifically, in support of its position, the Office cites to Voskuil's paragraphs [0014] and [0015].

These sections of Voskuil simply describe the notion that a client application can have the ability to be extended by third party software components, and a specific mechanism for effecting discovery on a end user computer. There is nothing in these excerpts or anywhere else in Voskuil that discloses or suggests the specific subject matter of this claim.

Accordingly, for at least these reasons, the Office has failed to establish a *prima facie* case of obviousness.

Claims 30-39 depend from claim 29 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited

1 features which, in combination with those recited in claim 29, are neither disclosed
2 nor suggested in the references of record, either singly or in combination with one
3 another.

4
5 **Claims 40-47**

6 **Claim 40** recites a data structure embodied on a computer-readable
7 medium comprising [emphasis added]:

- 8
- 9 • a first sub-structure indicative of a software extension that is to be
10 incorporated in a software application program;
 - 11 • one or more second sub-structures associated with the first sub-
12 structure and indicating *feature types* that are added by the extension
13 to the application program; and
 - 14 • one or more third sub-structures associated with the one or more
15 second sub-structures and indicating features of an associated
16 *feature type* that are added by the extension.

17
18 In making out the rejection of this claim, the Office argues essentially that
19 Mutschler discloses all of the claim's subject matter except for the recited
20 "software application program". For this feature, the Office relies on Voskuil.

21 In addressing the application of Mutschler, the Office first argues that
22 Mutschler discloses the first sub-structure by virtue of using XML. The Office
23 apparently takes the position that because XML allows designers to create their
24 own customized tags for transmitting data between applications and organizations,
25 that somehow this meets the subject matter of this claim. The Office is wrong.
Mutschler either discloses a data structure having the recited features or it doesn't.
In this case, Mutschler does not disclose subject matter that meets the first recited
claim element.

1 Continuing, the Office argues that because the definition of XML allows
2 for extensions to be extended, this in some way meets the recited “one or more
3 second sub-structures”. Applicant is at a loss to understand the Office’s logic.
4 Mutschler either discloses a data structure having the recited features or it doesn’t.
5 In this case, Mutschler does not disclose subject matter that meets the first or
6 second recited claim elements.

7 Continuing, the Office argues that Mutschler meets the third recited feature
8 and cites to Mutschler’s discussion of its DTD. Applicant is again at a loss to
9 understand the Office’s logic. Mutschler either discloses a data structure having
10 the recited features or it doesn’t. In this case, Mutschler does not disclose subject
11 matter that meets the first, second or third recited claim elements.

12 In its reliance on Voskuil, the Office argues that it discloses a “software
13 application program” and its combination with Mutschler would render the subject
14 matter of this claim obvious because “XML is an object based hierarchical
15 programming construct, that by the very definition is extensible, therefore any
16 software extension may be further extended to provide additional functionality.”
17 See, Office Action, page 15.

18 This motivation falls far short of the particular findings that are required to
19 support a *prima facie* case of obviousness. This alone (not to mention in
20 combination with the Office’s misinterpretation of Mutschler), fails to establish a
21 *prima facie* case of obviousness.

22 **Claims 41-47** depend from claim 40 and are allowable as depending from
23 an allowable base claim. These claims are also allowable for their own recited
24 features which, in combination with those recited in claim 40, are neither disclosed
25

1 nor suggested in the references of record, either singly or in combination with one
2 another.

3
4 **Claims 48-53**

5 **Claim 48** recites a method of delivering software via a network comprising
6 [emphasis added]:

- 7
- 8 • navigating to a network site that maintains at least one software
application program; and
 - 9 • downloading a ***software application program*** from the network site,
10 the application program comprising multiple different functionalities
11 that can assist a user in accomplishing different tasks, the ***software
application program*** being configured to be extended with software
12 extensions that are deliverable via a network and are described by at
least one network-deliverable file.
- 13

14 In making out the rejection of this claim, the Office cites to column 2, lines
15 19-22 and 27-31, of Mutschler, reproduced above, and to Voskuil's paragraphs
16 [0014] and [0029], also reproduced above. The Office then argues that this
17 claim's subject matter is obvious over Mutschler in view of Voskuil. As pointed
18 out above, Mutschler has nothing to do with software extensions as that term is
19 utilized in this claim. Rather, Mutschler pertains to tools that allow data
20 interchange among software tools. The excerpts of Mutschler cited by the Office
21 have nothing to do with software extensions or their delivery.

22 In addition, paragraph [0014] in Voskuil simply describes the notion that a
23 client application is extendible. Further, paragraph [0029] discusses the notion
24 that a client interface supports an extension interface, that an extension module
25 can access a file system to create, read and write files, and that the extension

1 modules can receive instructions from a profile server. Nowhere do these excerpts
2 disclose or suggest a network-deliverable file that *describes* a software extension,
3 as that notion is defined in Applicant's specification. If the Office is of a different
4 mind, then Applicant respectfully requests the Office to specifically point to what
5 it considers to be a network-deliverable file that describes a software extension.

6 Accordingly, the Office has not established a *prima facie* case of
7 obviousness and this claim is allowable.

8 **Claims 49-53** depend from claim 48 and are allowable as depending from
9 an allowable base claim. These claims are also allowable for their own recited
10 features which, in combination with those recited in claim 48, are neither disclosed
11 nor suggested in the references of record, either singly or in combination with one
12 another.

13 14 **Claim 54**

15 **Claim 54** recites one or more computer-readable media having computer-
16 readable instructions thereon which, when executed by a computer, cause the
17 computer to [emphasis added]:

- 18
- 19 • navigate to a network site that maintains at least one software
application program;
 - 20 • download a software application program comprising multiple
different functionalities that can assist a user in accomplishing
21 different tasks, the software application program being configured to
be extended with software extensions that are deliverable via the
22 network and described by at least one network-deliverable file; and
 - 23 • extend the software application program by adding at least one
extension to the application program, the extension being added by
24 using a link to navigate to a different network site that hosts one or
25 more files that describe the extension, and extension files that are

1 used to implement the extension and downloading the one or more
2 files and the extension files to a client.

3 In making out the rejection of this claim, the Office argues that its subject
4 matter is rendered obvious in view of Mutschler and Voskuil. In support of its
5 position, the Office cites to Mutschler's column 4, lines 24-39 and column 5, lines
6 18-19. These excerpts of Mutschler simply describe Mutschler's solution for
7 generating formats to allow development tools to share information. Specifically,
8 as instructed by Mutschler, its solution provides application developers with a
9 common language for specifying transfer syntax for a development language that
10 allows visualizing, constructing and documenting distributed objects and business
11 models. These excerpts do not, as the Office contends, disclose all of the subject
12 matter of this claim except for software delivery over a network. More
13 specifically, Mutschler has nothing to do with, nor does it pertain to software
14 extensions as utilized in this claim and described in Applicant's specification. In
15 addition.

16 The Office then relies on Voskuil for the notion of software delivery over a
17 network, citing to paragraph [0014] for support. As a motivation to combine
18 Mutschler and Voskuil, the Office argues essentially that the motivation would
19 exist because "Mutschler's code for object models or code for software extensions,
20 as disclosed by Voskuil, is merely code."

21 The Office has failed to establish a *prima facie* case of obviousness for at
22 least the following reasons. First, Mutschler does not disclose software extensions
23 as that term is utilized in the claim and described in Applicant's specification.
24 Second, the Office's stated motivation—because object models and software
25

1 extensions are merely code—falls far short of providing a convincing line of
2 reasoning that is stated with particularity.

3 Accordingly, the Office has failed to establish a *prima facie* case of
4 obviousness and this claim is allowable.

5
6 **Claims 55-62**

7 **Claim 55** recites a method comprising [emphasis added]:

- 8
- 9 • accessing a Web site through which one or more software extensions
 - 10 • receiving at least one file that describes at least one software
11 extension using a hierarchical language that describes the software
12 extension's logical attachment to a software application program;
 - 13 • receiving one or more software extension files; and
 - 14 • installing the one or more software extension files based, at least in
15 part, on the description contained in said at least one file.

16 The Office contends that the subject matter of this claim, except for
17 software delivery over a network, is disclosed in column 4, lines 21-39, and
18 column 6, lines 11-16 and 29-49 of Mutschler. The Office then relies on Voskuil
19 for the notion that software can be delivered over a network.

20 Applicant can find no disclosure in Mutschler that describes receiving at
21 least one file that describes at least one software extension using a hierarchical
22 language that describes the software extension's logical attachment to a software
23 application program, receiving one or more software extension files, and installing
24 the extension file(s) based, at least in part, on the description contained in the file
25 that describes the software extension. As such, the Office has failed to establish a
prima facie case of obviousness and this claim is allowable.

1 **Claims 56-62** depend from claim 55 and are allowable as depending from
2 an allowable base claim. These claims are also allowable for their own recited
3 features which, in combination with those recited in claim 55, are neither disclosed
4 nor suggested in the references of record, either singly or in combination with one
5 another.

6
7 **Claim 63**

8 **Claim 63** recites a method comprising [emphasis added]:

- 9
- 10 • describing one or more software extensions using one or more
11 extensible markup language (XML) files, the extensions being
12 configured for incorporation in a software program executing on a
13 client;
 - 14 • associating the one or more XML files with one or more associated
15 extension files that are useable to provide a program functionality;
16 and
 - 17 • storing the XML files and associated extension files in a network-
18 accessible location;
 - 19 • said acts of describing and associating being configured to provide
20 *software* for delivery over the network.

21 In making out the rejection of this claim, the Office argues that the first two
22 elements of this claim are disclosed in Mutschler's column 4, lines 21-39. The
23 Office is mistaken. This excerpt of Mutschler simply describes Mutschler's
24 solution for allowing development tools to share information. Nowhere can
25 Applicant find a disclosure that anticipates the first two elements of this claim.

26 Further, the Office argues that the last recited element of the claim is
27 disclosed in Mutschler's column 2, line 20-22. This excerpt is simply one of
28 Mutschler's stated objects of the invention. It simply states that an object of

1 Mutschler's invention is to allow developers of distributed systems the ability to
2 share object models and *other metadata* over a network. Applicant is at a loss to
3 understand or appreciate how such a statement in Mutschler can be construed to
4 disclose anticipating the acts of "describing" and "associating" when, in fact,
5 Mutschler fails to disclose describing one or more software extension using one or
6 more XML files, and associating the one or more XML files with one or more
7 associated extension files.

8 Accordingly, for at least this reason, the Office has failed to establish a
9 *prima facie* case of obviousness and this claim is allowable. Given this, the
10 Office's reliance on Voskuil is misplaced and adds nothing of significance. In
11 addition, the Office's stated motivation to combine these references—because
12 each reference discloses "merely code" falls short of the convincing line of
13 reasoning that is necessary to support a *prima facie* case of obviousness.
14 Accordingly, for this additional reason, the Office has failed to establish a *prima*
15 *facie* case of obviousness.

16 17 **Claims 64-65**

18 **Claim 64** recites a network site comprising:

- 19
- 20 • one or more software extension files configured to be incorporated
21 into a software application program, the software extension files
22 being configured to allow delivery of software via a network; and
 - 23 • one or more files associated with the one or more software extension
24 files and describing the extension files, the one or more files
25 describing a logical attachment of the one or more software
extension files to the software application program.

1 The Office contends that the subject matter of this claim, except for the
2 delivery of the software extensions, is disclosed by Mutschler in column 4, lines
3 21-39, and column 6, lines 11-16 and 29-49. However, as discussed above,
4 Mutschler does not teach, in these excerpts or anywhere else, software extensions
5 as that term is utilized in the claim and described in Applicant's specification. In
6 addition, Mutschler fails to disclose the notion of any file that describes an
7 extension file, where such file describes a logical attachment of a software
8 extension file to a software application program. Given this, the Office has failed
9 to establish a *prima facie* case of obviousness.

10 As such, the Office's reliance on Voskuil is misplaced. In addition, the
11 Office's stated motivation to combine these references—because each reference
12 discloses “merely code” falls short of the convincing line of reasoning that is
13 necessary to support a *prima facie* case of obviousness. Accordingly, for this
14 additional reason, the Office has failed to establish a *prima facie* case of
15 obviousness.

16 **Claim 65** depends from claim 64 and is allowable as depending from an
17 allowable base claim. This claim is also allowable for its own recited features
18 which, in combination with those recited in claim 64, are neither disclosed nor
19 suggested in the references of record, either singly or in combination with one
20 another.

21 **Claims 66-69**

22 **Claim 66** recites a method of managing network-based software extensions
23 comprising [emphasis added]:
24
25

- grouping multiple software extension descriptions in a catalog in a network-accessible location to enable delivery of *software* via a network;
- accessing the network-accessible location; and
- using the catalog to update a software extension that is resident on a computing device.

In making out the rejection of this claim, the Office cites to column 5, lines 16-23, and column 6, lines 11-12 and 22-36 of Mutschler for the proposition that Mutschler discloses all of the subject matter of this claim, except for software delivery. However, as discussed above, Mutschler does not teach, in these excerpts or anywhere else, grouping *multiple software extension descriptions* in a catalog in a network-accessible location to enable delivery of software via a network, or using the catalog to *update a software extension* that is resident on a computing device. For at least this reason, the Office has failed to establish a *prima facie* case of obviousness and this claim is allowable.

As such, the Office's reliance on Voskuil is misplaced. In addition, the Office's stated motivation to combine these references—because each reference discloses “merely code” falls short of the convincing line of reasoning that is necessary to support a *prima facie* case of obviousness. Accordingly, for this additional reason, the Office has failed to establish a *prima facie* case of obviousness.

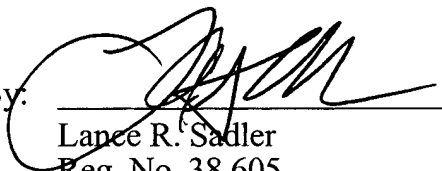
Claims 67-69 depend from claim 66 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 66, are neither disclosed nor suggested in the references of record, either singly or in combination with one another.

1
2 **Conclusion**

3 Applicant has studied the reference cited by the Office and has sincerely
4 attempted to describe how the claimed subject matter patentably distinguishes over
5 this reference. Applicant submits that all of the claims are in condition for
6 allowance and respectfully requests that the Office pass the application along to
7 issuance. If the Office's next anticipated action is to be anything other than
8 issuance of a Notice of Allowability, Applicant respectfully requests a telephone
9 call for the purpose of scheduling an interview.

10 Respectfully Submitted,

11
12 Dated: 5/2/05

13 By: 
14 Lance R. Sadler
15 Reg. No. 38,605
16 (509) 324-9256 ext. 226
17
18
19
20
21
22
23
24
25